一种具有平衡标签预测能力的在线哈希算法

何 硕，谢 良†

(武汉理工大学 理学院，武汉 430070)

摘 要：针对传统离线哈希算法训练模型耗时、占用内存大和不易更新模型的问题，以及现实图像集的标签存在大量损失的现象，提出了一种能够平衡标签预测的在线哈希算法(balanced label prediction for online hashing，BLPOH)。BLPOH 通过标签预测模块生成预测标签，并融合残缺的真实标签，能够有效缓解因标签损失导致的模型性能下降。观察到标签存在分布不平衡现象，提出标签类别相似性平衡算法并应用于标签预测模块，提升标签预测的准确性。将旧数据的信息加入哈希函数的在线更新过程，提升模型对旧数据的兼容性。通过在两个广泛使用的数据集上进行实验，并和一些当前先进的算法进行对比，结果证实了 BLPOH 的优越性。

关键词：在线哈希；多标签；标签预测；图像检索

中图分类号：TP391 doi: 10.19734/j.issn.1001-3695.2022.02.0088

Online hash algorithm with balanced label prediction ability

He Shuo, Xie Liang†

(*School of Faculty of Science*, *Wuhan University of Technology*, *Wuhan* 430070, *China*)

Abstract: Aiming at the problems that the traditional offline hashing algorithm is time-consuming to train the model, occupies a large amount of memory and is difficult to update the model, and there is a large loss of labels in real image sets, this paper proposed a balanced label prediction for online hashing algorithm(BLPOH) . BLPOH generated predicted labels through the label prediction module, and fused the incomplete real labels, which can effectively alleviate the performance degradation of the model caused by label loss. It observed that there is an imbalance in the distribution of labels, then proposed a label category similarity balance algorithm to apply to the label prediction module to improve the accuracy of label prediction. It added the information of old data to the online update process of the hash function to improve the compatibility of the model to the old data. By conducting experiments on two widely used datasets and comparing with some state-of-the-art algorithms, the results demonstrate the superiority of BLPOH.

Key words: online hashing; multi label; label prediction; image retrieval

0 引言

哈希算法由于其高效的检索能力和占用存储低的特点而在图像检索领域一直备受关注。哈希算法把高维的图像特征映射到一个低维的二值空间[1]，得到一连串紧凑的哈希码，这串哈希码保留了图像间的相似性信息[2]。正是由于哈希码的这些特性，使得图像检索任务可以在一个低维空间高效地进行，而哈希码的质量也就成为影响图像检索性能的关键因素。

研究者们致力于学习一个优秀的哈希函数，得到高质量的哈希码。Gionis 等人[3]提出局部敏感哈希算法(Locality-Sensitive Hashing, LSH)。LSH 是一种无监督哈希算法，通过将原始数据随机映射到不同的哈希桶中，然后经过符号函数得到哈希码。Weiss 等人[4]提出谱哈希算法(Spectral Hashing, SH)。SH 以图像分割的思想解决图像特征的编码问题。Wang 等人[5]提出基于半监督学习的图像哈希检索算法 (Semi-Supervised Hashing, SSH)。Liu 等人[6]提出图哈希算法(Hashing with Graphs, AGH)。AGH 有很多思想源于 SH，不同的是 AGH 通过数据聚类中心和数据样本点之间的近邻图近似样本点之间的近邻图。Shen 等人[7]提出基于有监督的哈希算法(Supervised Discrete Hashing, SDH)。SDH 通过离散循环下降算法(Cyclic Coordinate Descent, DCC)按位求解哈希码，避免松弛化求解引起的次优问题。

然而，以上提出的哈希算法都是通过离线学习[8]的方式一次性从所有数据中学习哈希函数。面对越来越庞大的图像数据集，这些离线哈希算法在学习过程中不仅占用大量的内存空间，而且十分耗时。不仅如此，每当新数据到来，离线哈希算法为了更新模型不得不重新进行训练。面对这样的应用场景，在线哈希算法(Online Hashing)能够很好地解决这些问题。在线哈希算法流式地读取数据[9]，每次只从一个数据流中训练并更新模型，占用空间小，训练模型快，并且易于更新模型。近年来，研究者们提出了多种在线哈希算法。作为代表的在线哈希算法包括在线核哈希[10](Online Kernel Hashing, OKH)，在线监督哈希[11](Online Supervised Hashing, OSH)，自适应哈希[12](Adaptive Hashing, AdaptHash)，在线草图哈希[13](Sketching Hashing, SketchHash)，具有共同信息的在线哈希[14](Online Hashing with Mutual Information, MIHASH)和平衡相似性在线离散哈希[15] (Balanced Similarity for Online Discrete Hashing, BSODH)，高效更新哈希码的在线哈希[16](Online Hashing with Efficient Updating of Binary Codes, OHWEU)，哈达玛矩阵引导的在线哈希[17](Hadamard Matrix Online Hashing, HMOH) 以及通过哈达玛码书学习的监督在线哈希[18](Online Hashing via Hadamard Codebook, HCOH)。

虽然这些在线哈希算法解决了离线哈希算法在巨大数据集上较难训练和更新模型的问题，但是它们仍然存在一些缺点。现实可用的图像数据集总会由于图像标注[19]误差造成图像标签的缺失，然而当前的在线哈希算法都默认用到的数据

收稿日期：2022-02-25；**修回日期：**2022-04-26

作者简介：何硕(1997-)，男，河南驻马店人，硕士研究生，主要研究方向为图像哈希；谢良(1987-)，男(通信作者)，副教授，博士，主要研究方向为多媒体大数据检索、语义分析与跨模态分析以及机器学习的应用(whutxl@hotmail.com).

集的标签是完整的，这造成模型精度的下降。以 BSODH 为例，只考虑多标签数据的情况下，以两幅图像是否共享至少一个标签来判断它们是否相似[20]，由于图像标签并不完整，导致构造的图像标签相似性矩阵不准确，最终造成模型精度的损失。

本文提出了一种能够平衡标签预测的在线哈希算法(BLPOH)。引入在线哈希算法，避免离线哈希算法在巨大图像数据集上训练耗时，占用内存大和加入新数据需要重新训练的问题。考虑到现实图像标签存在缺失的问题，加入标签预测[21]模块，通过图像相似性和标签类别相似性预测近邻图像的标签，以对抗图像标签缺失引起的性能损失。由于多数标签类别是不相似的，相似的只有少数，所以，标签类别存在不平衡现象。因此，在标签预测模块设计了一个标签类别相似性平衡算法，提升标签预测的准确性。此外，本文巧妙地根据汉明距离[22]的退化形式度量新旧数据之间哈希码和标签的距离，使旧数据的信息参与哈希函数的在线更新。本文在 MIR Flickr25k 和 NUS-WIDE 两个数据集上进行广泛的实验，并和一些当前先进的算法进行对比以证明 BLPOH 的优越性。

本文的主要贡献如下：

a) 本文引入在线哈希算法，解决了离线哈希在巨大数据集训练和更新模型难的问题。并通过加入标签预测模块，缓解了现实图像数据标签缺失引起的性能下降问题。

b) 本文考虑到标签类别分布不平衡问题，在标签预测模块设计标签类别相似性平衡算法，提升预测的准确性。

c) 本文通过汉明距离的退化形式建立新旧数据之间哈希码和标签的相似性对等关系，让旧数据的信息监督哈希函数的在线更新，解决了在线哈希在旧数据上性能下降的问题，提升哈希函数对旧数据的兼容性。

1 算法框架

图 1 展示了 BLPOH 的整体框架。每当数据流中有新数据到来时，通过找出图像的 K 近似最近邻计算图像相似性矩阵，并把该矩阵应用到标签预测当中。然后，通过预测的标签计算图像标签相似性矩阵，并利用该矩阵与旧数据建立关联，使得旧数据的信息也参与到哈希函数的在线更新过程。最后，通过哈希函数计算出图像数据的哈希码。

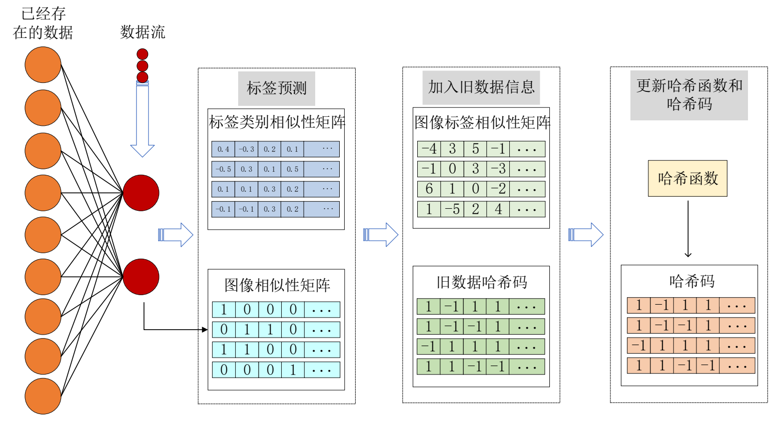


图 1 算法框架图

Fig. 1 Algorithm framework diagram

1.1 问题定义

给出一个多标签数据集，$X=[x_1,\dots,x_n]^T\in\mathbb{R}^{n\times d}$ 表示图像特征集，$L=[l_1,\dots,l_n]^T\in\mathbb{N}^{n\times u}$ 表示标签集，其中 $x_i\in\mathbb{R}^d$ 是第 i 个图像实例，它对应的标签是 $l_i\in\mathbb{N}^u$，n 是实例的个数，u 是标签类别数量，d 是图像特征的维度。哈希函数把图像特征数据映射为 $B=[b_1,\dots,b_n]^T\in\{-1,+1\}^{n\times k}$，其中 b_i 是图像实例 x_i 对应的二进制哈希码向量，k 是哈希码的位数。一般情况下，哈希函数都采用如 BSODH 的线性哈希映射，为了适合本文的数据处理方式，将哈希函数定义为

$$B=F(X)=\operatorname{sgn}(XW) \tag{1}$$

其中，W 是投影矩阵。$\operatorname{sgn}(x)$ 是符号函数，当变量 $x>0$ 时返回 1，否则返回-1。

将在线哈希算法的数据流表示为 X_s^t 和 X_e^t，其中 $X_s^t=[x_{s1}^t,\dots,x_{sn_t}^t]^T\in\mathbb{R}^{n_t\times d}$ 表示 t 时刻到来的新数据，$X_e^t=[x_{e1}^t,\dots,x_{em_t}^t]^T\in\mathbb{R}^{m_t\times d}$ 表示 t 时刻已经存在的旧数据，以及它们在 t 时刻对应的真实标签矩阵 $L_s^t=[l_{s1}^t,\dots,l_{sn_t}^t]^T\in\mathbb{N}^{n_t\times u}$ 和 $L_e^t=[l_{e1}^t,\dots,l_{em_t}^t]^T\in\mathbb{N}^{m_t\times u}$，$n_t$ 表示 t 时刻数据流中的数据量，m_t 表示 t 时刻已经存在的数据量。对应地，把 t 时刻数据流中的新数据对应的哈希码矩阵表示为 $B_s^t=\operatorname{sgn}(X_s^tW^t)=[b_{s1}^t,\dots,b_{sn_t}^t]^T\in\{-1,+1\}^{n_t\times k}$，把 t 时刻已经存在的旧数据对应的哈希码矩阵表示为 $B_e^t=\operatorname{sgn}(X_e^tW^t)=[b_{e1}^t,\dots,b_{em_t}^t]^T\in\{-1,+1\}^{m_t\times k}$。在本文的在线哈希算法中，随着新数据 X_s^t 的到来，以在线的方式更新 t 时刻的哈希码投影矩阵 W^t。把 t 时刻数据流对应的预测标签矩阵表示为 $Y_s^t=[y_{s1}^t,\dots,y_{sn_t}^t]^T\in\{-1,+1\}^{n_t\times u}$，把已经存在的旧数据对应的预测标签矩阵表示为 $Y_e^t=[y_{e1}^t,\dots,y_{em_t}^t]^T\in\{-1,+1\}^{m_t\times u}$，-1 表示没有该标签，1 表示拥有该标签。

表 1 介绍了本文主要的变量及其定义。

表 1 关键变量的介绍

Tab. 1 Introduction to key variables

变量	定义
X_s^t	t 时刻的数据流
X_e^t	t 时刻已经存在的旧数据
Y_s^t	t 时刻数据流 X_s^t 对应的预测标签矩阵
Y_e^t	t 时刻旧数据 X_e^t 对应的预测标签矩阵
B_s^t	t 时刻数据流 X_s^t 对应的哈希码矩阵
B_e^t	t 时刻旧数据 X_e^t 对应的哈希码矩阵
W^t	t 时刻哈希函数对应的投影矩阵
V^t	t 时刻 X_s^t 和 Y_s^t 间的投影矩阵

1.2 模型框架

1.2.1 标签预测及标签类别相似性平衡

正如前面提到的，现实可用的图像数据集总会由于标注误差导致数据集的标签存在缺失。考虑到这种情况，本文通过建立标签预测模块来对抗标签缺失引起的模型精度下降问题。受到文献[21]的启发，在假设标签类别相似矩阵 Q_s 已知的情况下，一个图像实例的标签可以由该图像实例的最近邻近似地表示。类似地，一个图像实例的标签也可以由该标签的最近邻近似地表示。建立如文献[21]的标签预测正则化项，得到 t 时刻的关系表达公式：

$$\min_{Y_s^t}\|Y_s^t-S_i^tY_s^tQ_s^t\|_F^2\quad s.t.\ Y_s^t\in\{-1,1\}^{n_t\times u} \tag{2}$$

其中，S_i^t 是 t 时刻通过图像的 K 近似最近邻计算得到的图像实例间的相似性矩阵。$\|\cdot\|_F$ 表示 Frobenius 范数。

和文献[21]不同的是，本文创新性的提出标签类别相似

性平衡算法。标签类别相似性平衡算法基于标签类别分布不平衡现象：绝大多数标签对是不相似的，少数标签对是相似的。为了解决这个问题，本文将标签类别相似矩阵分为 Q_{s1} 和 Q_{s2} 两部分，它们分别表示相似标签对和非相似标签对，对于 Q_{s1}，给它分配一个较小的权重，而给 Q_{s2} 分配一个较大的权重，以此让标签类别相似性向标签对不相似的方向偏移。对于没有添加标签类别相似性平衡算法的模型，为了达到和添加该算法的模型同等精度，需要迭代更多的次数，导致在相同迭代次数下预测的标签的准确性和预测的收敛速度都较为低下。将标签类别相似性平衡算法的公式定义如下：

$$\tilde{Q}_{ij}^t=\begin{cases}\eta_s Q_{ij}^t, & Q_{ij}^t>0\\ \eta_d Q_{ij}^t, & Q_{ij}^t<0\end{cases} \tag{3}$$

其中，η_s 表示较小的权重，η_d 表示较大的权重。

1.2.2 映射关系及其损失函数的定义

此外，建立 X 到预测标签 Y 的线性映射关系，Y 以 XV 表示，建立 t 时刻 $X_s^tV^t$ 和 Y_s^t 的 F 范数损失：

$$\min_{V^t}\|X_s^tV^t-Y_s^t\|_F^2 \tag{4}$$

哈希算法的目的是学习一个哈希函数，根据式(1)，最小化线性哈希函数 F 和 X_s^t 对应哈希码 B_s^t 之间的误差，建立公式：

$$\min_{W^t}\|X_s^tW^t-B_s^t\|_F^2 \tag{5}$$

1.2.3 旧数据的信息参与哈希函数的在线更新

以汉明距离度量两个维度相同的向量之间的距离。如果两个数据越相似，则它们对应的二进制哈希码之间的汉明距离越小，反之越大。在训练新数据和更新模型的同时，得到的新模型应该同样适用于旧数据。因此，从数据流中训练模型的同时，应该把旧数据的信息考虑在内。考虑到如果数据 x_i 和 x_j 是相似的，那么它们就有相似的标签矩阵，而且由哈希函数的特性可知，哈希码保留了数据间的相似性关系，因此能够通过哈希码和标签数据建立新数据和旧数据之间的相似性对等关系。考虑到，当哈希码以-1 和 1 表示时，那么哈希码 b_i 和 b_j 之间的汉明距离 h 就可以表示为

$$h=1/2(k-<b_i,b_j>) \tag{6}$$

其中，$<b_i,b_j>$ 表示 b_i 和 b_j 的内积。

同样，对于某个标签类别，以-1 和 1 分别标识数据没有该标签和拥有该标签。最小化两者间的 F 范数损失得：

$$\min_{B_s^t,B_e^t}\|B_s^tB_e^{tT}-S_l^t\|_F^2 \quad s.t.\ B_s^t\in\{-1,1\}^{n_t\times k},B_e^t\in\{-1,1\}^{m_t\times k} \tag{7}$$

其中，S_l^t 是标签相似性矩阵。

为了有效利用存在损失的真实标签信息，在计算标签相似性矩阵 S_l^t 时，将存留的真实标签信息附加到对应的预测标签上，即将真实的标签矩阵中为 1 处的值加到对应位置的预测标签矩阵中，充分利用已经存在的真实标签信息，进一步提升模型的精度。因为只需要利用标签存在的信息，所以把 L_s^t 和 L_e^t 中值为-1 的位置替换为 0，为 1 的位置不变，分别用 A_s^t 和 A_e^t 表示。S_l^t 的计算定义如下：

$$S_l^t=k(Y_s^t+A_s^t)(Y_e^t+A_e^t)^T/u \tag{8}$$

综上，建立 BLPOH 的模型公式：

$$\begin{aligned}\min_{Q_s^t,B_s^t,B_e^t,W^t,V^t,Y_s^t}&\|Y_s^t-S_i^tY_s^tQ_s^t\|_F^2+\\ &\alpha\|B_s^tB_e^{tT}-S_l^t\|_F^2+\beta\|X_s^tW^t-B_s^t\|_F^2+\\ &\varsigma\|X_s^tV^t-Y_s^t\|_F^2+\lambda(\|W^t\|_F^2+\|V^t\|_F^2)\\ s.t.\ &B_s^t\in\{-1,1\}^{n_t\times k},B_e^t\in\{-1,1\}^{m_t\times k},Y_s^t\in\{-1,1\}^{n_t\times u}\end{aligned} \tag{9}$$

1.3　算法优化

由于存在二进制约束问题，所以采用迭代的方式进行求解，在更新一个变量时固定其他变量，把其他变量当作常量处理。

1) 更新 W^t：固定 Q_s^t，B_s^t，B_e^t，V^t 和 Y_s^t，然后学习 W^t。求解 W^t 的子问题是：

$$\min_{W^t}\beta\|X_s^tW^t-B_s^t\|_F^2+\lambda(\|W^t\|_F^2) \tag{10}$$

因此，可以得到式(10)的闭式解：

$$W^t=\beta^t(\beta^tX_s^{tT}X_s^t+\lambda^tI)^{-1}X_s^{tT}B_s^t \tag{11}$$

其中，I 是一个 $d\times d$ 的单位矩阵。

2) 更新 V^t：和更新 W^t 一样，固定 Q_s^t，B_s^t，B_e^t，W^t 和 Y_s^t，然后学习 V^t。求解的子问题是：

$$\min_{V^t}\varsigma\|X_s^tV^t-Y_s^t\|_F^2+\lambda(\|V^t\|_F^2) \tag{12}$$

因此，可以得到 V^t 的闭式解：

$$V^t=\varsigma^t(\varsigma^tX_s^{tT}X_s^t+\lambda^tI)^{-1}X_s^{tT}Y_s^t \tag{13}$$

3) 更新 B_e^t：同样固定其他所有变量，更新 B_e^t。求解子问题是：

$$\min_{B_e^t}\alpha\|B_s^tB_e^{tT}-S_l^t\|_F^2 \quad s.t.\ B_e^t\in\{-1,1\}^{m_t\times k} \tag{14}$$

类似文献[23]，式(14)又可以退化为求解：

$$\min_{B_e^t}\|B_s^tB_e^{tT}-S_l^t\|_1 \quad s.t.\ B_e^t\in\{-1,1\}^{m_t\times k} \tag{15}$$

求解式(15)，得

$$B_e^t=\mathrm{sgn}(S_l^{tT}B_s^t)\in\{-1,1\}^{m_t\times k} \tag{16}$$

4) 更新 B_s^t：固定其他所有变量，学习 B_s^t。求解子问题为

$$\min_{B_s^t}\alpha\|B_s^tB_e^{tT}-S_l^t\|_F^2+\beta\|X_s^tW^t-B_s^t\|_F^2 \quad s.t.\ B_s^t\in\{-1,1\}^{n_t\times k} \tag{17}$$

展开式(17)为

$$\begin{aligned}\min_{B_s^t}&\alpha(\|B_e^tB_s^{tT}\|_F^2+\|S_l^t\|_F^2-2tr(B_s^tB_e^{tT}S_l^{tT}))+\\ &\beta(\|B_s^t\|_F^2+\|W^{tT}X_s^{tT}\|_F^2-2tr(B_s^tW^{tT}X_s^{tT}))\\ s.t.\ &B_s^t\in\{-1,1\}^{n_t\times k}\end{aligned} \tag{18}$$

式(18)可以简化为

$$\min_{B_s^t}\alpha\|B_e^tB_s^{tT}\|_F^2-2tr(\alpha B_s^tB_e^{tT}S_l^{tT}+\beta B_s^tW^{tT}X_s^{tT}) \quad s.t.\ B_s^t\in\{-1,1\}^{n_t\times k} \tag{19}$$

和 BSODH 类似，使 $P=\alpha B_e^{tT}S_l^{tT}+\beta W^{tT}X_s^{tT}$，得到：

$$\min_{B_s^t}\alpha\underbrace{\|B_s^tB_e^{tT}\|_F^2}_{term\mathrm{I}}-\underbrace{2tr(B_s^tP)}_{term\mathrm{II}} \quad s.t.\ B_s^t\in\{-1,1\}^{n_t\times k} \tag{20}$$

其中，$term\mathrm{I}=\tilde{b}_{sr}^t\tilde{b}_{er}^{t\,T}+\tilde{B}_s^t\tilde{B}_e^{tT}$，$term\mathrm{II}=\tilde{b}_{sr}^t\tilde{p}_r+\tilde{B}_s^t\tilde{P}$，$\tilde{b}_{sr}^t$，$\tilde{b}_{er}^t$，$\tilde{p}_r$ 分别代表代表 B_s^t，B_e^t，P 的第 r 行，$\tilde{B}_s^t$，$\tilde{B}_e^t$，$\tilde{P}$ 分别代表除了 $\tilde{b}_{sr}^t$，$\tilde{b}_{er}^t$，$\tilde{p}_r$ 余下的行。

式(20)转换为

$$\min_{\tilde{b}_{sr}^t}\alpha\|\tilde{b}_{sr}^t\tilde{b}_{er}^{t\,T}+\tilde{B}_s^t\tilde{B}_e^{tT}\|_F^2-2tr(\tilde{b}_{sr}^t\tilde{p}_r)-2tr(\tilde{B}_s^t\tilde{P}) \quad s.t.\ \tilde{b}_{sr}^t\in\{-1,1\}^{1\times k} \tag{21}$$

求解式(21)，得

$$\tilde{b}_{sr}^t=\mathrm{sgn}(\tilde{p}_r^{\,T}-\alpha\tilde{B}_s^t\tilde{B}_e^{tT}\tilde{b}_{er}^t)\in\{-1,1\}^{1\times k} \tag{22}$$

通过迭代求解每一行，得到最终的 B_s^t。

5) 更新 Y_s^t：固定其他所有变量，学习 Y_s^t。求解子问题为

$$\min_{Y_s^t}\|Y_s^t-S_i^tY_s^tQ_s^t\|_F^2+\varsigma\|X_s^tV^t-Y_s^t\|_F^2 \quad s.t.\ Y_s^t\in\{-1,1\}^{n_t\times u} \tag{23}$$

将式(23)转换为普通的线性方程：

$$(M^TM+\varsigma I_V)Vec(Y_s^t)=Vec(\varsigma X_s^tV^t) \tag{24}$$

其中，$M=I_V-L_S$，$L_S=Q_s^{tT}\otimes S_i^t$，$\otimes$ 是 Kronecker 积，I_V 是一个 $un_t\times un_t$ 的单位矩阵。

6) 更新 Q_s^t：固定其他所有变量，更新 Q_s^t。过程和求解 Y_s^t 类似，得

$$(I_L\otimes((S_i^tY_s^t)^T(S_i^tY_s^t)))Vec(Q_s^t)=Vec((S_i^tY_s^t)^TY_s^t) \tag{25}$$

其中，I_L 是一个 $uu\times uu$ 的单位矩阵。

算法 1　BLPOH 的描述

输入：数据集 X，哈希码位数 k，数据流大小 n_t，参数 α，β，ς，λ，η_s/η_d 和图像的最近邻个数 K。

输出：W 和 B，其中 $W=W^t$，$B=\mathrm{sgn}(XW)$。

a) 对数据集 X 进行归一化处理，并根据 n_t 从 X 中切分出 X_s^t，随机

初始化 W^t 和 V^t，初始化 Q_s^t 为一个单位矩阵，并计算初始的 B_s^t 和 Y_s^t。

b) 根据 K 计算 X_s^t 中数据间的相似性矩阵 S_i^t。

c) 根据式(11)和(13)分别更新 W^t 和 V^t。

d) 根据式(16)更新 B_e^t。

e) 根据式(22)迭代更新 B_s^t。

f) 根据式(24)和(25)分别更新 Y_s^t 和 Q_s^t。

g) 设置 $X_e^t=[X_e^t;X_s^t]$，$B_e^t=[B_e^t;B_s^t]$，$Y_e^t=[Y_e^t;Y_s^t]$，并从 X 中切分出下一次的 X_s^t。

h) 重复步骤 b)~g)。

2　实验与分析

2.1　数据集

在这篇文章中，本文在两个公开可用的数据集上进行实验。为了更准确地模拟图像标签损失的情况，需要构造确定标签存在损失的数据集。首先，从数据集中筛选出具有特定标签数量以上的数据，建立实验所需的图像数据集和对应的标签集。然后，本文使用卷积神经网络 VGG-16 对图像数据进行特征提取，提取的特征维度为 4096。同时，对图像对应的标签集进行编码，建立标签矩阵，为了模型计算相似度的需要，如果一个图像包含该标签，就把该标签所在位置标识为 1，否则为-1。最后，像文献[21]那样考虑不完整标签率，随机丢弃训练数据{0%, 20%, 40%, 60%, 80%}的标签，分别建立这五种状态下的子数据集。

MIR Flickr25k 数据集包含 25000 张图像数据和 24 个标签类别，本文筛选具有 3 个及 3 个以上标签数量的数据，总共筛选出 17568 张图像。在提取完图像特征和建立标签矩阵之后，再以{0%, 20%, 40%, 60%, 80%}的比率随机丢弃标签，模拟数据标签丢失标签的情况。实验中，除了分析数据库的数量对实验结果影响的实验，对于其他的实验，本文从挑选出的 17568 个图像数据中随机选择 12400 个作为实验的数据集，其中 10000 个作为数据库用作训练，剩下的 2400 个图像数据用作查询集。

NUS-WIDE 数据集包含 269648 张图像数据和 81 个标签类别，本文选择出现最多的 21 个类别作为标签。处理过程和 MIR Flickr25k 类似，总共筛选出 57073 张图像数据，并从中随机选择 12100 个作为实验的数据集，其中 10000 个作为数据库用作训练，剩下的 2100 个图像数据用作查询集。

2.2　基准方法与评估指标

本文以一些前沿的在线哈希算法作为基准方法，包括 BSODH，MIHASH，OKH，AdaptHash，SketchHash，HMOH 和 OHWEU。实验时，BLPOH 和对比算法对比使用的数据集和数据集的标签损失率等设定完全一致，对于每个对比算法的参数，专门设定使其得到最好的效果。实验设备的操作系统为 Windows10，CPU 为 R5-3600，内存为 32G。当数据集是 MIR Flickr25k 时，实验设置参数 $\alpha=\beta=0.8$，$\varsigma=0.9$，$\lambda=0.6$，流数据量为 200，最近邻 $K=10$，$\eta_s=0.4$，$\eta_d=1$。当数据集是 NUS-WIDE 时，实验设置参数 $\alpha=\beta=\varsigma=0.9$，$\lambda=0.1$，流数据量为 200，最近邻 $K=12$，$\eta_s=0.2$，$\eta_d=1$。本文使用平均精度均值(Mean Average Precision, MAP)作为度量的标准来评估这些算法的性能。

2.3　实验结果

2.3.1 MAP 对比

表 2~4 和图 2 展示了 BLPOH 和对比算法在数据集为 MIR Flickr25k 和 NUS-WIDE，并且标签损失率分别为{0, 20%, 40%, 60%, 80%}时不同哈希码位数下的 MAP 结果对比和变化曲线。

如表 2 所示，在数据集 MIR Flickr25K 的不同标签损失率下，BLPOH 的 MAP 始终高于其他所有对比算法。其中，HMOH 和 OHWEU 的整体效果和 BLPOH 较为接近，AdaptHash 的整体效果相比最差。在 NUS-WIDE 数据集下，当标签损失率为 0 的时候，BLPOH 的 MAP 高于除 HMOH 外的其他对比算法，当标签损失率继续上升时，BLPOH 的 MAP 超过包括 HMOH 在内的所有对比算法，证明了 BLPOH 的优越性能。从和 HMOH 在数据集 NUS-WIDE 下的结果中可以看出，当没有标签损失时，HMOH 的 MAP 高于 BLPOH，当标签损失率上升到 20%及以后时，BLPOH 的 MAP 超过了 HMOH，证明了 BLPOH 对抗标签损失的有效性。在实际应用中，标签的损失程度可能会非常严重，BLPOH 更适用于实际情况。

图 2 给出了 BLPOH 和对比算法在数据集 MIR Flickr25k 和 NUS-WIDE 不同哈希码位数下的 MAP 随标签损失率上升的变化图。从中可以看出，在不同数据集和不同哈希码位数的情况下，BLPOH 的 MAP 的下降更加平缓，且多数情况下总能保持更高的水平。

2.3.2 关键参数灵敏度分析

本小节分析标签类别相似性权重参数 η_s 和 η_d 对实验结果的影响。在实验中，设置 $\eta_d=1$，并通过改变 η_s/η_d 的大小来分析它们对实验结果的影响。从图 3(a)可以看出，在数据集 MIR Flickr25k 上，当 η_s/η_d 的值介于(0.3, 0.5)时，MAP 出现最大值，当 η_s/η_d 在 0.8 以后 MAP 大幅下降，当 $\eta_s/\eta_d=1$ 时 MAP 的值最低，此时表示的是没有使用标签类别相似性平衡算法的 MAP，最终本文选取 $\eta_s=0.4$，$\eta_d=1$ 作为实际参数值。如图 3(b)所示，在数据集 NUS-WIDE 上，$\eta_s/\eta_d=0.2$ 时 MAP 最高，当 $\eta_s/\eta_d=1$ 时 MAP 最低，最终选取 $\eta_s=0.2$，$\eta_d=1$ 作为实际参数值。从标签类别相似性权重比值在两种数据集上的表现可以看出，没有使用标签类别相似性平衡的 MAP 总是低于使用标签类别相似性平衡的，这证明了标签类别相似性平衡对 BLPOH 的有效性。

表 2　在 MIR Flickr25k 和 NUS-WIDE 上且哈希码位数为 32 时的 MAP

Tab. 2　The MAP of MIR Flickr25k and NUS-WIDE when the hash bits is 32

方法	MIR Flickr25k					NUS-WIDE				
	0	20%	40%	60%	80%	0	20%	40%	60%	80%
BSODH	0.716	0.714	0.636	0.503	0.260	0.843	0.769	0.662	0.496	0.282
MIHASH	0.836	0.736	0.553	0.370	0.207	0.850	0.714	0.534	0.347	0.201
OKH	0.797	0.692	0.532	0.363	0.199	0.855	0.719	0.535	0.357	0.204
AdaptHash	0.783	0.672	0.523	0.352	0.205	0.856	0.719	0.540	0.359	0.203
SketchHash	0.803	0.697	0.544	0.373	0.216	0.869	0.736	0.553	0.369	0.211
HMOH	0.837	0.767	0.653	0.523	0.267	**0.883**	0.799	0.683	0.530	0.279
OHWEU	0.789	0.737	0.647	0.506	0.306	0.869	0.792	0.680	0.520	0.292
BLPOH	**0.872**	**0.812**	**0.714**	**0.562**	**0.336**	0.881	**0.802**	**0.703**	**0.532**	**0.303**

表 3　在 MIR Flickr25k 和 NUS-WIDE 上且哈希码位数为 64 时的 MAP

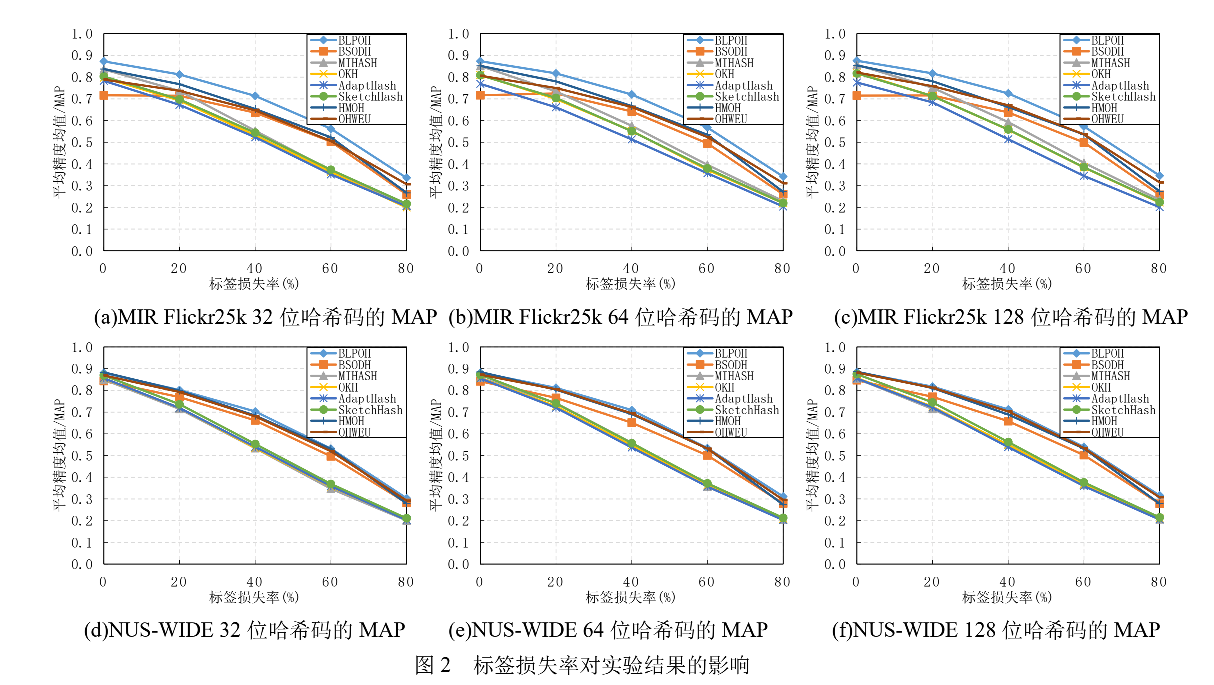
Tab. 3　The MAP of MIR Flickr25k and NUS-WIDE when the hash bits is 64

方法	MIR Flickr25k					NUS-WIDE				
	0	20%	40%	60%	80%	0	20%	40%	60%	80%
BSODH	0.717	0.725	0.642	0.494	0.261	0.842	0.765	0.651	0.501	0.279
MIHASH	0.851	0.734	0.576	0.396	0.229	0.858	0.741	0.559	0.356	0.205
OKH	0.810	0.699	0.552	0.373	0.220	0.856	0.726	0.546	0.362	0.208
AdaptHash	0.769	0.661	0.513	0.357	0.204	0.855	0.720	0.537	0.357	0.203
SketchHash	0.809	0.704	0.551	0.379	0.220	0.872	0.740	0.557	0.372	0.213
HMOH	0.852	0.781	0.667	0.534	0.272	**0.885**	0.803	0.691	0.532	0.274
OHWEU	0.806	0.750	0.663	0.524	0.311	0.873	0.805	0.694	0.531	0.296
BLPOH	**0.873**	**0.817**	**0.720**	**0.568**	**0.342**	0.882	**0.812**	**0.709**	**0.534**	**0.311**

表 4　在 MIR Flickr25k 和 NUS-WIDE 上在哈希码位数为 128 时的 MAP

Tab. 4　The MAP of MIR Flickr25k and NUS-WIDE when the hash bits is 128

方法	MIR Flickr25k					NUS-WIDE				
	0	20%	40%	60%	80%	0	20%	40%	60%	80%
BSODH	0.715	0.716	0.637	0.499	0.258	0.847	0.771	0.658	0.502	0.278
MIHASH	0.857	0.751	0.593	0.405	0.235	0.854	0.714	0.550	0.366	0.206
OKH	0.820	0.713	0.558	0.384	0.222	0.857	0.725	0.549	0.366	0.210
AdaptHash	0.775	0.684	0.514	0.345	0.201	0.855	0.721	0.539	0.359	0.205
SketchHash	0.817	0.713	0.559	0.385	0.224	0.875	0.745	0.562	0.376	0.215
HMOH	0.854	0.781	0.662	0.536	0.274	**0.886**	0.811	0.689	0.533	0.278
OHWEU	0.822	0.759	0.670	0.537	0.315	0.882	0.812	0.703	0.534	0.308
BLPOH	**0.876**	**0.817**	**0.725**	**0.573**	**0.346**	0.885	**0.817**	**0.712**	**0.539**	**0.316**

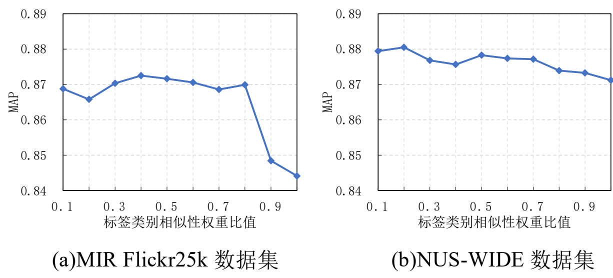


(a)MIR Flickr25k 32 位哈希码的 MAP　(b)MIR Flickr25k 64 位哈希码的 MAP　(c)MIR Flickr25k 128 位哈希码的 MAP

(d)NUS-WIDE 32 位哈希码的 MAP　(e)NUS-WIDE 64 位哈希码的 MAP　(f)NUS-WIDE 128 位哈希码的 MAP

图 2　标签损失率对实验结果的影响

Fig. 2　The effect of label loss rate on results



(a)MIR Flickr25k 数据集　(b)NUS-WIDE 数据集

图 3　标签类别相似性权重比值对实验结果的影响

Fig. 3　The effect of label category similarity weight ratio on results

2.3.3 数据库样本数量对实验结果的影响

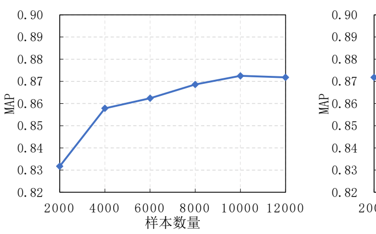
BLPOH 是一种基于在线哈希的改进算法，为了测试BLPOH 作为在线哈希算法的检索性能，本文分析了 BLPOH 的实验结果随数据库样本增加的变化情况。设置 MIR Flickr25k 和 NUS-WIDE 查询集的数量分别为 2400 和 2100，剩下的用作训练。如图 4(a)所示，数据集为 MIR Flickr25k，当数据库的样本数量为 2000 时，BLPOH 的 MAP 较其他五种情况最差，当样本数量在 4000 以后，MAP 逐渐增加，当样本数量为 10000 时，BLPOH 的 MAP 基本趋于稳定。如图 4(b)所示，BLPOH 在 NUS-WIDE 数据集下的 MAP 从数据库样本数量为 2000 开始逐渐上升，在样本数量为 8000 时趋于稳定。

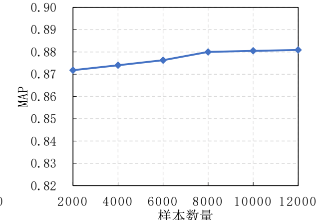
2.3.4 标签类别相似性平衡算法对实验结果的影响

在算法优化过程中，本文把优化预测标签矩阵 Y_s^t 和标签类

别相似性矩阵 Q_s 转换为求解对应的线性方程组，即式(23)和(24)。在实际编程计算中，本文使用预条件共轭梯度法(Preconditional Conjugate Gradient, PCG)对式(23)和(24)进行求解。

本节通过 PCG 的收敛位置来分析标签类别相似性平衡对 BLPOH 的影响。以 NUS-WIDE 数据集下的标签预测和求解标签类别相似性矩阵为例。如图 5(a)所示，在每次数据流到来并进行标签预测时，使用标签类别相似性平衡算法总能比未使用标签类别相似性平衡算法收敛的更快，且每次收敛的位置变化较小，表现更稳定。同样地，如图 5(b)，在求解标签类别相似性矩阵时，使用标签类别相似性平衡算法相比未使用标签类别相似性平衡算法的表现也相对稳定一些。

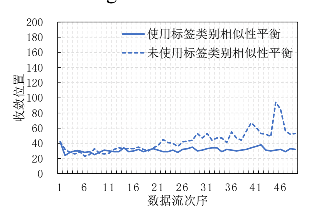


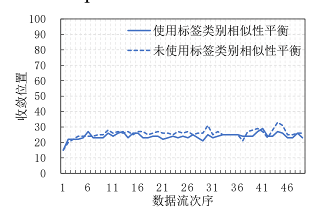


(a)MIR Flickr25k 数据集　　(b)NUS-WIDE 数据集

图 4　不同样本数量对实验结果的影响

Fig. 4　The effect of different sample sizes on results





(a)标签预测　　(b)求解标签类别相似性矩阵

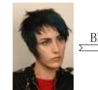
图 5　标签类别相似性平衡对不同求解过程收敛位置的影响

Fig. 5　Influence of label categories similarity balance on convergence location of different solution processes

2.3.5 图像实例的检索效果

为了展现 BLPOH 的实际检索效果，本文和 BSODH 在数据集 MIR Flickr25K 中的图像实例检索效果进行对比。

实验时，从 MIR Flickr25K 数据集中选取一张图像，根据相同条件下训练出的模型返回前 5 张最相似的图像。结果如图 6 所示，图像下的文本为该图像的标签。根据 BLPOH 训练的模型返回了 5 张人像，而根据 BSODH 训练的模型返回了 4 张人像和一张动物的图像。BLPOH 的检索效果明显优于 BSODH。











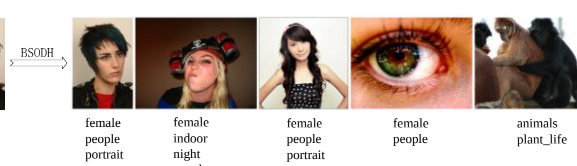


图 6　blpoh 和 bsodh 在 mir flickr25k 中的实例检索

Fig. 6　Instance retrieval of the algorithm in this paper and bsodh in mir flickr25k

3　结束语

本文提出了改进的在线哈希算法，流式地读取数据，以在线的方式训练和更新模型，使得算法模型不受制于数据集的大小，并使模型契合当前的数据信息，使模型具有良好的实时性。并且，让旧数据的信息监督哈希函数的在线更新，提升哈希函数对旧数据的兼容性。此外，考虑到现实图像数据标签存在缺失的问题，加入标签预测模块，并且新颖地提出标签类别相似性平衡算法，指导标签的预测，使得预测的标签更加精确，也提升了图像标签相似度的准确性，进一步提升模型精度。此外，本文的标签预测模块不仅能让模型能够很好地对抗标签缺失，并且在标签损失率不断上升时，也能够使模型的性能以较平缓的速度下降，在标签存在缺失的在线图像检索任务中具有良好的应用前景。然而，本文的标签预测模型比较单一，模型的健壮性并不能令人非常满意，所以在今后的工作中考虑加入集成学习的思想提升模型的健壮性。

参考文献：

[1] 郭一村，陈华辉．在线哈希算法研究综述 [J]．计算机应用，2021, 41 (04): 1106-1112. (Guo Yicun, Chen Huahui. Survey on online hashing algorithm [J]. Journal of Computer Applications, 2021, 41 (04): 1106-1112.)

[2] Lin Guosheng, Shen Chunhua, Shi Qinfeng, *et al.* Fast supervised hashing with decision trees for high-dimensional data [C]// Proc of the 27th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2014: 1963-1970.

[3] Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing [C]// Proc of the 25th Very Large Data Base Conference. San Francisco, CA: Morgan Kaufmann Publishers, 1999: 518-529.

[4] Weiss Y, Fergus R, Torralba A. Multidimensional spectral hashing [C]// Proc of the 12th European Conference on Computer Vision. Berlin: Springer Verlag, 2012: 340-353.

[5] Wang Jun, Kumar S, Chang S F. Semi-supervised hashing for large-scale search [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2012, 34 (12): 2393-2406.

[6] Liu Wei, Mu Cun, Kumar S, *et al.* Discrete graph hashing [C]// Proc of the 27th International Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2014: 3419-3427.

[7] Shen Fumin, Shen Chunhua, Liu Wei, *et al.* Supervised discrete hashing [C]// Proc of the 28th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2015: 37-45.

[8] Bombara G, Belta C. Offline and online learning of signal temporal logic formulae using decision trees [J]. ACM Trans on Cyber-Physical Systems, 2021, 5 (3): 1-23.

[9] Lu Xu, Zhu Lei, Cheng Zhiyong, *et al.* Flexible online multi-modal hashing for large-scale multimedia retrieval [C]// Proc of the 27th ACM International Conference on Multimedia. New York: Association for Computing Machinery, 2019: 1129-1137.

[10] Huang Longkai, Yang Qiang, Zheng Weishi. Online hashing [C]// Proc of the 23rd International Joint Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press, 2013: 1422-1428.

[11] Cakir F, Bargal S A, Sclaroff S. Online supervised hashing [J]. Computer Vision and Image Understanding, 2017, 156: 162-173.

[12] Cakir F, Sclaroff S. Adaptive hashing for fast similarity search [C]// Proc of the 15th IEEE International Conference on Computer Vision. NW Washington, DC: IEEE Computer Society, 2015: 1044-1052.

[13] Leng Cong, Wu Jiaxiang, Cheng Jian, *et al.* Online sketching hashing [C]// Proc of the 28th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2015: 2503-2511.

[14] Cakir F, He Kun, Adel Bargal S, *et al.* Mihash: Online hashing with

mutual information [C]// Proc of the 16th IEEE International Conference on Computer Vision. NW Washington, DC: IEEE Computer Society, 2017: 437-445.
[15] Lin Mingbao, Ji Rongrong, Liu Hong, *et al.* Towards optimal discrete online hashing with balanced similarity [C]// Proc of the 33rd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2019: 8722-8729.
[16] Weng Zhenyu, Zhu Yuesheng. Online hashing with efficient updating of binary codes [C]// Proc of the 34th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2020: 12354-12361.
[17] Lin Mingbao, Ji Rongrong, Liu Hong, *et al.* Hadamard matrix guided online hashing [J]. International Journal of Computer Vision, 2020, 128 (8): 2279-2306.
[18] Lin Mingbao, Ji Rongrong, Liu Hong, *et al.* Supervised online hashing via hadamard codebook learning [C]// Proc of the 26th ACM International Conference on Multimedia. New York: Association for Computing Machinery, 2018: 1635-1643.
[19] 王琳，张素兰，杨海峰. 基于 CNN 和加权贝叶斯的最近邻图像标注方法 [J]. 计算机技术与发展, 2021, 31 (10): 63-69. (Wang Lin, Zhang Sulan, Yang Haifeng. A nearest neighbor image annotation method based on CNN and weighted bayesian [J]. Computer Technology and Development, 2021, 31 (10): 63-69.)
[20] Jiang Qingyuan, Li Wujun. Deep cross-modal hashing [C]// Proc of the 30th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2017: 3232-3240.
[21] Dong Haochen, Li Yufeng, Zhou Zhihua. Learning from semi-supervised weak-label data [C]// Proc of the 32nd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2018: 2926-2933.
[22] Norouzi M, Punjani A, Fleet D J. Fast search in hamming space with multi-index hashing [C]// Proc of the 25th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2012: 3108-3115.
[23] Kang Wangcheng, Li Wujun, Zhou Zhihua. Column sampling based discrete supervised hashing [C]// Proc of the 30th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2016: 1230-1236.